Autonomous Runtime Systems

Dorian Arnold (U. of New Mexico) and Samuel Gutierrez (Los Alamos National Laboratory)
darnold@cs.unm.edu

Motivation: Many of the applications and tools to be deployed on future exascale systems, comprised of millions of cores and billion-way parallelism, will be characterized by dynamic computation and communication workloads. For example, many scientific applications, such as molecular dynamics codes, particle in cell codes, fluid dynamics codes and general finite element methods on irregular objects codes [14, 12, 4], compute on irregular data structures and with dynamic communication patterns. These irregular data structures, which include unbalanced trees and graphs and unstructured grids lead to dynamic scheduling and load balancing challenges [15]. Similarily, tools for performance analysis, debugging and resource management dynamically vary their dataflow workloads and the analyses on these flows based on the specific task at hand. Such dynamic communication patterns generally lead to different execution phases; across these phases workload characteristics and requirements often change. These include changes in communication groups and patterns, payload sizes and computational complexity. Such dynamicism suggests dynamic adjustments to the way resources are mapped to the application or even changes in the number of resources allocated to the application. Additionally, expectedly high failure rates on next generation systems suggest that as components fail and recover, runtime systems must be able to dynamically shrink and grow allocations.

Our Position and Approach: We believe that *autonomous operation* will be a key component in the efficient, scalable and robust operation of future HPC systems. An autonomous systems is generic, self monitoring, self (re)configuring, self healing and self optimizing. Such operation will be necessary to manage the volumes of data to be analyzed and the large numbers of components to be managed on future systems: management strategies that rely on on human involvment simply will not work.

Our primary goal is to develop scalable mechanisms that allow exascale applications and tools to execute efficiently and effectively without reliance on human intervention and expertise. We also will investigate new runtime system strategies necessary to support dynamic application workloads and characteristics and dynamic resource demands and availabilities. More specifically, our aims are:

- To design and evaluate mechanisms for scalable, autonomous middleware for communication and data aggregation. These mechanisms will be responsible for
  - sensing and monitoring dynamic environmental characteristics including available resources and data workload intensity,
  - detecting when functional or performance problems arise,
  - deciding what actions if any should be taken to address the problems,
  - and instantiating the actions.
- To design and evaluate a framework based on autonomous components for efficient deployment of irregular applications at extreme scales; and
- To engineer practical, portable solutions of our ideas for empirical evaluation and broader use by the community.

Our approach has two primary prongs:

1. We propose to develop scalable, dynamic autonomous mechanisms for identifying and instantiating near optimal resource allocations and task/resource mappings. These mechanisms will monitor health and performance characteristics, diagnose performance and functional problems, evaluate remedial actions and their costs and benefits, and instantiate prescribed remedies.
2. We propose a new resource management paradigm that improves resource usage by dynamically migrating resources back and forth between user application sessions and the general resource pool depending upon user demand, dynamic workload properties and resource availability. Our autonomous components will serve as the basis for our new runtime system by dynamically determining exactly how much resources are needed by application sessions and how to efficiently use the allocated resources.

The challenges in this work include designing and developing techniques for detecting, diagnosing and correcting performance failures that are accurate, effective and lightweight enough to be run alongside the primary application or tool without overly consuming its resources. To date, we have begun preliminary investigations of both the modeling and performance prediction aspects of tree-based overlay network performance and understanding the impact of virtual to physical topology mappings. We have also built all the necessary components to effect topology changes in the MRNet TBŌN prototype – this work was done as part of a project to make MRNet robust to process and node failures.

Related Work: Generally, there is little research in high-performance computing systems that focus on the use of autonomous concepts for addressing performance issues. The most closely related project is the Charm++-based Adaptive MPI (AMPI) project [8] which uses process and thread migration to achieve dynamic load balancing. AMPI is a implementation of virtual MPI processes, which are transparently mapped onto physical processors. By abstracting the location of processes away from the physical implementation, AMPI is able to dynamically adjust processor load balancing as well as the ratio of communication to computation. AMPI has a rich set of mechanisms, including MPI process virtualization and migration, to support dynamic runtime behavior. The fundamental difference between AMPI's strategy and our approach is that task are re-arranged based on automatic CPU load balancing strategies. We target automatic diagnosis and correction of communication performance problems. Additionally, we will explore topology reconfigurations – akin to "connection migrations" – as well as task migration.

There are many motivating studies and research projects that demonstrate the impact of topology aware mappings for improving the performance of HPC applications, for example, recent studies for IBM and Cray supercomputers [7, 2, 3, 5, 9, 11, 6]. A variety of mapping strategies have been explored from simulated annealing to graph contraction and even genetic algorithms. In general, previous approaches have concentrated on latency optimizations often in application and system specific ways. Furthermore, some of these techniques assume that application instances are allocated contiguous portions of a system. This proposed research additionally focuses on communication bandwidth and network contention issues.

Assessment:

- *Challenges addressed*: This work addresses the challenges of dynamic, scalable and efficient runtime systems for extreme scale applications and tools.
- *Maturity*: The most mature work in this area is Charm++, which has demonstrated the utility of dynamic load balancing strategies in contexts similar to ours. We also have demonstrated the feasibility of dynamic overlay network reconfigurations albeit in the context of fault tolerance. We have begun preliminary work on the tree-based performance models, which show the ability to accurately model and predict performance. To the best of our knowledge, the additional areas of autonomous overaly network operation are unexplored.
- *Uniqueness:* Exascale systems pose the unique challenge of unprecedented system scales that dramatically exacerbate the need for lightweight fault-tolerance techniques.
- *Novelty*: To the best of our knowledge, there is no work that attempts completely autonomous overlay network management to respond to both performance and functional component failures.
- *Applicability*: This line of research could be useful in all high-performance computing areas, from petascale on up. In general, the approaches are also useful beyond the HPC domain.
- *Effort*: This is a multi-year effort that entails both modeling and conceptualization analysis in the early years and infrastructure development and empirical validations in the out years.

References

[1] A. Bhatele. *Automating Topology Aware Mapping for Supercomputers*. PhD thesis, University of Illinois at Urbana Champaign, 2010.

[2] A. Bhatele and L. Kale. Application-specific topology-aware mapping for three dimensional topologies. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1 –8, april 2008.

[3] A. Bhatelé, L. V. Kalé, and S. Kumar. Dynamic topology aware load balancing algorithms for molecular dynamics applications. In *Proceedings of the 23rd international conference on Supercomputing*, ICS '09, pages 110–116, New York, NY, USA, 2009. ACM.

[4] R. Das, D. J. Mavriplis, J. Saltz, S. Gupta, and R. Ponnusamy. The design and implementation of a parallel unstructured euler solver using software primitives. In *AIAA Journal*, 1992.

[5] K. Davis, A. Hoisie, G. Johnson, D. J. Kerbyson, M. Lang, S. Pakin, and F. Petrini. A performance and scalability analysis of the bluegene/l architecture. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, SC '04, pages 41–, Washington, DC, USA, 2004. IEEE Computer Society.

[6] C. V. S. T. B. Deborah Weisser, Nick Nystrom and J. Urbanic. Optimizing Job Placement on the Cray XT3. In *48th Cray User Group Proceedings*, 2006.

[7] P. H. E. L. J. C. S. G. Bhanot, A. Gara and R. Walkup. Optimizing task layout on the blue gene/l supercomputer. *IBM Journal of Research and Development*, (49(2/3)):489–500, 2005.

[8] C. Huang, O. Lawlor, and L. V. Kale. Adaptive mpi. In *16th International Workshop on Languages and Compilers for Parallel Computing (LCPC 03)*, pages 306–322, 2003.

[9] E. Huedo, M. Prieto, I. Llorente, and F. Tirado. Impact of pe mapping on cray t3e message-passing performance. In A. Bode, T. Ludwig, W. Karl, and R. Wismller, editors, *Euro-Par 2000 Parallel Processing*, volume 1900 of *Lecture Notes in Computer Science*, pages 199–207. Springer Berlin / Heidelberg, 2000.

[10] J. Hursey, R. L. Graham, G. Bronevetsky, D. Buntinas, H. Pritchard, and D. G. Solt. Run-through stabilization: An MPI proposal for process fault tolerance. In *EuroMPI 2011: Proceedings of the 18th EuroMPI Conference*, Santorini, Greece, September 2011.

[11] S. H. Langer, B. Still, P.-T. Bremer, D. Hinkel, B. Langdon, J. A. Levine, and E. Williams. Cielo full-system simulations of multi-beam laser-plasma interaction in NIF experiments. In *CUG 2011, Proceedings of the 53rd Cray User Group Meeting*, Fairbanks, AK, May 2011. Cray User Group, Inc.

[12] B. Moon and J. Saltz. Adaptive runtime support for direct simulation monte carlo methods on distributed memory architectures. In *Scalable High-Performance Computing Conference, 1994., Proceedings of the*, pages 176 –183, may 1994.

[13] M. Salehie and L. Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 4:14:1–14:42, May 2009.

[14] Y. shin Hwang, R. Das, J. H. Saltz, and B. Brooks. Parallelizing molecular dynamics programs for distributed memory machines. *IEEE Computational Science and Engineering*, 2:18–29, 1995.

[15] K. A. Yelick, S. Chakrabarti, E. Deprit, J. Jones, A. Krishnamurthy, and C. po Wen. Data structures for irregular applications. In *DIMACS Workshop on Parallel Algorithms for Unstructured and Dynamic Problems*, 1993.